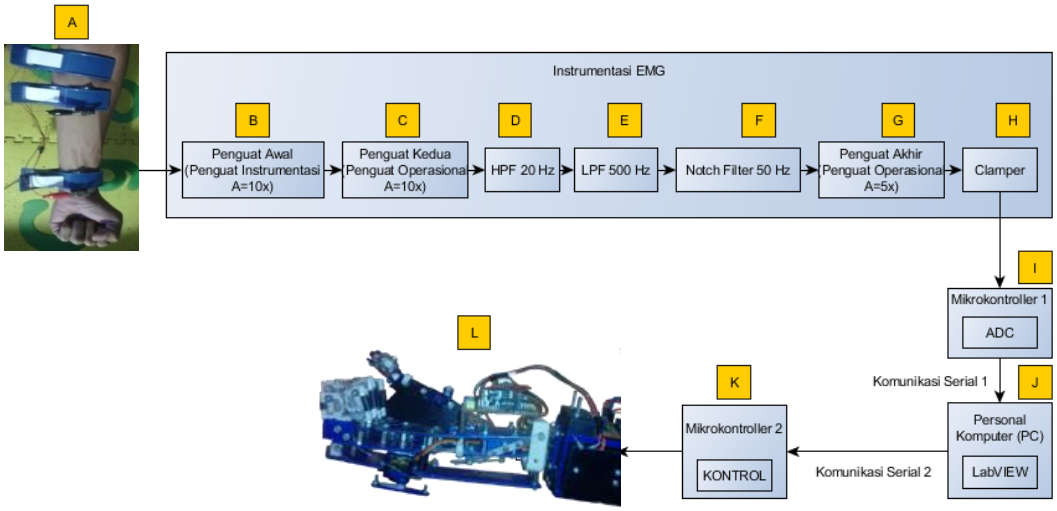
BAB III

PERANCANGAN SISTEM PENGENDALIAN PROSTETIK TANGAN ROBOTIK MELALUI PENDETEKSIAN SINYAL EMG

Perancangan merupakan proses menuangkan ide atau gagasan berdasarkan teori-teori dasar yang mendukung, kemudian merealisasikannya ke dalam bentuk sebuah alat/benda yang akan dirancang. Pada prinsipnya perancangan yang baik itu dilakukan secara sistematik atau teratur, sehingga mempermudah proses pembuatan alat serta analisanya.

3.1 Perancangan Sistem

Pada dasarnya sebelum merancang suatu alat secara sistematik dibutuhkan suatu blok diagram sistem. Pada **Gambar 3.1** di bawah ini adalah blok diagram dari "Sistem Pengendalian Prostetik Tangan Robotik Melalui Pendeteksian Sinyal EMG" yang dirancang.



Gambar 3.1. Blok Diagram Sistem

Pada blok diagram pada **Gambar 3.1** di atas dua buah mikrokontroler digunakan untuk memisahkan proses pembacaan ADC dan pengontrolan gerakan tangan robot, karena pembacaan ADC membutuhkan eksekusi perintah yang terus menerus dilakukan, sehingga tidak boleh ada eksekusi perintah lain yang menyebabkan pembacaan ADC akan terganggu dan menjadi *noise* bagi sinyal EMG yang terbaca. Dan juga untuk mempercepat proses eksekusi perintah untuk mengontrol pergerakan tangan robot, karena tangan robot ini memiliki enam buah motor servo yang akan dikontrol derajat putarannya.

Adapun penjelasan dan prinsip kerja dari blok diagram sistem pada **Gambar 3.1** sendiri diuraikan sebagai berikut.

A. Sensor Elektroda

Elektroda adalah sensor yang mengubah energi dari sinyal otot manusia menjadi sinyal elektris. Otot manusia akan menghasilkan sinyal yang akan diterima oleh sensor elektroda dan mengubahnya ke sinyal elektris. Sinyal otot yang terbaca oleh sensor ini sangatlah kecil, yaitu sekitar 0-10 mV. Elektroda yang digunakan berjumlah tiga *leads*. Cara pemasangan elektroda ini secara unipolar, yaitu memasang satu *lead* elektroda pada otot yang akan diukur dan dua *lead* elektroda lainnya diletakan di bagian yang minim gerakan sebagai referensi.

B. Penguat Awal (Penguat Instrumentasi, A=10 Kali)

Untuk menguatkan sinyal yang memiliki amplitudo yang sangat kecil, maka dibutuhkan penguatan dengan rangkaian penguat instrumentasi. Penguatan yang dilakukan dilakukan secara bertingkat (*cascade*). Tujuan dari penguatan bertingkat ini adalah untuk menjaga parameter CMRR (*common mode*

rejection ratio). CMRR ini adalah perbandingan penguatan sinyal tegangan dengan penguatan sinyal *noise* yang dihasilkan, semakin besar nilai CMRR ini maka penguat instrumentasi tersebut semakin baik.

C. Penguat Kedua (Penguat Operasional, A=10 Kali)

Penguat ini adalah bagian dari *cascade* selanjutnya dari rangkaian penguat. Penguat ini bertujuan menjaga parameter CMRR tetap tinggi atau menjaga agar *noise* sinyal yang ikut terbaca tidak dikuatkan terlalu besar sebelum dilakukan proses *filter*.

D. HPF (*High Pass Filter*) 20 Hz

Pada umumnya sinyal otot yang terbaca rawan terhadap *noise* dari luar, seperti sinyal jantung dan sinyal dari jala listrik, maka itu dibutuhkan suatu *filter* untuk meredam *noise* yang ikut terbaca itu. Beberapa *filter* yang akan digunakan adalah HPF, LPF dan *Notch Filter*. HPF ini bertujuan untuk meloloskan frekuensi data yang lebih besar dan meredam frekuensi data yang lebih kecil daripada frekuensi *cut-off*-nya. Berdasarkan karakteristik dari sinyal EMG, frekuensi *cut-off* HPF yang digunakan adalah 20 Hz.

E. LPF (*Low Pass Filter*) 500 Hz

LPF ini bertujuan untuk meloloskan frekuensi data yang lebih kecil dan meredam frekuensi data yang lebih besar daripada frekuensi *cut-off*-nya. Berdasarkan karakteristik dari EMG, frekuensi *cut-off* LPF yang digunakan adalah 500 Hz.

F. *Notch Filter* 50 Hz

Notch Filter ini bertujuan untuk meredam frekuensi data yang sama dengan frekuensi *cut-off*-nya dan meloloskan frekuensi selainnya.

G. Penguat Akhir (Penguat Operasiona, A=5 Kali)

Setelah sinyal selesai dalam tahap *filter*, selanjutnya dilakukan proses *cascade* rangkaian penguat sinyal yang terakhir. Penguat ini bertujuan untuk menguatkan sinyal data yang telah di-*filter* sebelumnya agar masuk dalam *range* tegangan yang dapat dibaca oleh ADC pada mikrokontroler 1.

H. *Clamper* (*up to* 5 V)

Clamper ini bertujuan untuk menggeser *based line* dari sinyal data yang masuk agar nilainya berada di atas 0 (positif) dan agar sinyal data tidak terpotong. Karena sinyal data yang masuk kemungkinan masih memiliki amplitudo di bawah 0 (bernilai minus).

I. Mikrokontroler 1 (ADC)

Sinyal data yang masih dalam bentuk analog tersebut kemudian diubah menjadi sinyal digital melalui ADC (*analog to digital converter*). Sinyal digital dari mikrokontroler 1 ini selanjutnya akan dikirim ke *personal computer* (PC) melalui *port* serial.

J. *Personal Computer* (PC)

Pada *personal computer* (PC) ini akan dilakukan pemrosesan sinyal data digital yang dikirim oleh mikrokontroler 1 melalui *port* serial. Hasil pemrosesan sinyal digital tersebut akan dianalisa dengan melakukan proses *filter* digital, ekstraksi fitur dan klasifikasi sinyal EMG dengan menggunakan LabVIEW. Hasil ekstraksi fitur yang didapat selanjutnya akan dijadikan *input* untuk proses klasifikasi dengan sistem kecerdasan buatan yang akan digunakan, yaitu *artificial neural network* (ANN). ANN ini akan mengklasifikasikan sinyal EMG sesuai gerakan otot tangan yang dilakukan.

K. Mikrokontroler 2 (Kontrol Gerakan Tangan Robot)

Setelah mendapat data klasifikasi, p*ersonal computer* (PC) mengirim *char* sesuai dengan gerakan yang didapatkan sebagai perintah ke mikrokontroler 2 untuk mengontrol gerakan tangan robot melalui komunikasi serial.

L. Tangan Robot

Tangan robot ini digunakan sebagai devais prostetik yang pergerakannya dikendalikan oleh mikrokontroler 2 sesuai *char* yang diterima dari PC.

3.2 Pemilihan Komponen

Pemilihan komponen yang tepat akan berdampak pada kualitas kinerja alat dan efisiensi biaya. Menurut William D.C (1993) dan Sharon, dkk (1982), pemilihan komponen mengacu pada beberapa faktor, seperti sensitivitas, nilai ekonomi/harga, jangkauan dan respon waktu. Berikut adalah beberapa perbandingan komponen yang dipilih untuk perancangan rangkaian selanjutnya.

3.2.1 Pemilihan Jenis Komponen Penguat Instrumentasi

Pemilihan jenis penguat instrumentasi berhubungan dengan komponen yang digunakan untuk penguatan sinyal otot yang memiliki amplitudo sangat kecil.

Tabel 3.1. Perbandingan Penguat Instrumentasi

Tipe	**AD620**	**INA128**	**AD623**
Gain	1-10000x	1-1000x	1-1000x
Bandwidth	800 KHz	700 KHz	100 KHz
CMRR	93 dB-110 dB	100 dB-106 dB	100 dB-120 dB
Slew Rate	0,75 V/uS-1.2 V/uS	4 V/uS	0,3 V/uS
Supply Range	±2,3 V to ±18 V	±2,25 V to ±18 V	±2,5 V to ±6 V
Harga	Rp. 65.000, 00	Rp. 150.000, 00	Rp. 250.000, 00

Berdasarkan tabel perbandingan penguat instrumentasi di atas, penguat

instrumentasi dengan tipe AD620 dipilih karena beberapa kelebihan dibandingkan penguat instrumentasi lainnya, walaupun AD620 ini kalah dalam perbandingan CMRR, *slew rate*, dan *supply range*. AD620 memiliki beberapa kelebihan, yaitu *gain range* yang besar (10.000x), *bandwidth* yang lebih besar (800 KHz) dan harga yang lebih murah (Rp. 65.000, 00) dibandingkan tipe penguat instrumentasi yang lainnya.

3.2.2 Pemilihan Jenis Komponen Penguat Operasional

Pemilihan jenis penguat operasional ditujukan agar sinyal data analog yang terbaca dapat dikuatkan dengan baik sehingga sinyal datanya tidak berubah.

Tabel 3.2. Perbandingan Penguat Operasional

Tipe	**LF356**	**LM741**	**LF355**
Bandwidth	5 MHz	0,437 MHz-1,5 MHz	2,5 MHz
CMRR	80 dB-100 dB	70 dB-90 dB	80 dB-100 dB
Slew Rate	50 V/uS	0.5 V/uS	5 V/uS
Supply Range	±22 V	±22 V	±22 V
Harga	Rp. 20.000, 00	Rp 2.500, 00	Rp 15.000, 00

Berdasarkan tabel perbandingan penguat operasional di atas, penguat operasional dengan tipe LF356 memiliki beberapa kelebihan dibandingkan tipe yang lain, walaupun harganya lebih mahal daripada tipe yang lainnya. Beberapa kelebihan yang dapat dilihat adalah *bandwidth* yang lebih besar (5 MHz) dan *slew rate* yang lebih bagus (50 V/uS).

3.2.3 Pemilihan Jenis Komponen Mikrokontroler

Pemilihan jenis mikrokontroler akan digunakan untuk pembacaan sinyal data EMG dan untuk mengendalikan pergerakan dari tangan robot.

Tabel 3.3. Perbandingan Jenis Mikrokontroler

Tipe	**ATMega8535**	**AT89C51AC3**	**AT89S51**
Kecepatan Max	16 MHz	40 MHz	33 MHz
Jumlah Pin	40 pin	40 pin	40 pin/44 pin
Memori *Flash*	8 Kbytes	64 Kbytes	4 Kbytes
RAM	512 Bytes	256 Bytes	128 Bytes
Resolusi ADC	8 *Channel*, 10 bit	8 *Channel*, 10 bit	8 *Channel*, 10 bit
Operating Voltage	4,5 V-5,5 V	3,0 V-5,5 V	4,0 V-5,5 V
Harga	Rp. 50.000, 00	Rp. 267.000, 00	Rp. 11.000, 00

Berdasarkan tabel perbandingan mikrokontroler di atas, ATMega8535 memiliki kelebihan dalam kapasitas RAM (512 Bytes), walaupun kecepatan maksimumnya lebih rendah daripada mikrokontroler lainnya.

3.2.4 Pemilihan Tipe Motor Servo

Berikut adalah perbandingan beberapa tipe motor servo yang akan digunakan untuk perancangan mekanik tangan robot.

Tabel 3.4. Perbandingan Tipe Motor Servo

Tipe	**SG-90**	**MG-995**	**DSS-M15**
Berat	9 g	56 g	48 g
Torsi	1,6 kg/cm	9 kg/cm	10 kg/cm
Kecepatan	0,12 s/60deg	0,17 s/60deg	0,17 s/60deg
Operating Voltage	3,0 V-6,0 V	3,5 V-7,2 V	4,8 V-7,2V
Harga	Rp 55.000, 00	Rp 130.000, 00	Rp 265.000, 00

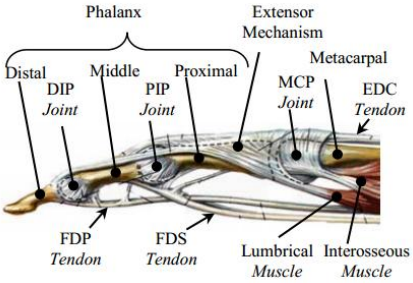
Berdasarkan tabel perbandingan tipe motor servo SG-90 dipilih sebagai aktuator penggerak jari-jari tangan robot karena lebih cepat (0,12 s/60*degree*), harga yang lebih murah (Rp 55.000,00), beratnya yang ringan (9 g) dan *operating voltage* yang lebih lebar. Dan untuk bagian lengan robot digunakan motor servo DSS-M15 yang memiliki torsi yang cukup besar, yaitu 10 kg/cm, motor servo ini dipilih juga karena menggunakan bahan metal *gear*.

3.3 Perancangan *Hardware*

Perancangan *hardware* yang akan dibuat meliputi perancangan mekanik tangan robot dan perancangan rangkaian yang terdiri atas instrumentasi EMG, sistem minimum mikrokontroler, rangkaian pengirim dan penerima data, dan rangkaian DC *Converter*.

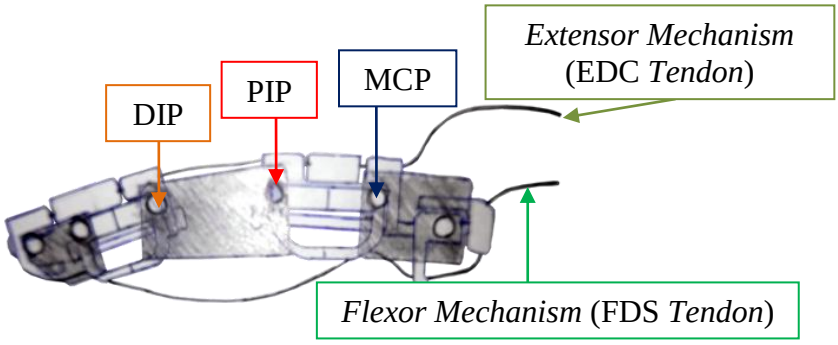
3.3.1 Perancangan Mekanik Tangan Robot

Dalam perancangan mekanik tangan robot ini yang paling difokuskan adalah pada pembuatan desain jari. Berdasarkan antatomi manusia, jari tangan manusia memiliki tiga sendi yang terdiri atas *metacarpophalangeal* (MCP), *proximales interphalangeal* (PIP) dan *distal interphalangeal* (DIP) yang ditunjukan pada **Gambar 3.2** berikut.



Gambar 3.2. Sendi Jari pada Tangan Manusia

Untuk pembuatan jari tangan robot dirancang mengikuti anatomi jari manusia, yaitu memiliki tiga sendi yang ditunjukan pada **Gambar 3.3** berikut.



Gambar 3.3. Desain Jari Tangan Robot

Untuk semua jari dirancang memiliki tiga sendi pergerakan, kecuali jari jempol yang hanya memiliki dua sendi saja. *Extensor mechanism* berfungsi untuk meluruskan jari tangan robot dan *flexor mechanism* berfungsi untuk membelokan jari tangan robot. Untuk perancangan *extensor mechanism* dan *flexor mechanism* dibuat menggunakan benang nilon yang ditarik oleh sebuah motor servo merk TowerPro tipe SG90 yang memiliki torsi 1,6 kg/cm.

Dan untuk bagian lengan menggunakan motor serpo tipe DSS-M15 yang memiliki torsi 10 kg/cm. Lengan robot ini dirancang dapat berputar 180^{o} untuk melakukan proses gerakan pronasi (menelungkapkan tangan) dan supinasi (menengadahkan tangan) pada lengan tersebut. Pada **Gambar 3.4** berikut adalah hasil perancangan tangan robot secara keseluruhan.



Gambar 3.4. Hasil Perancangan Keseluruhan Tangan Robot

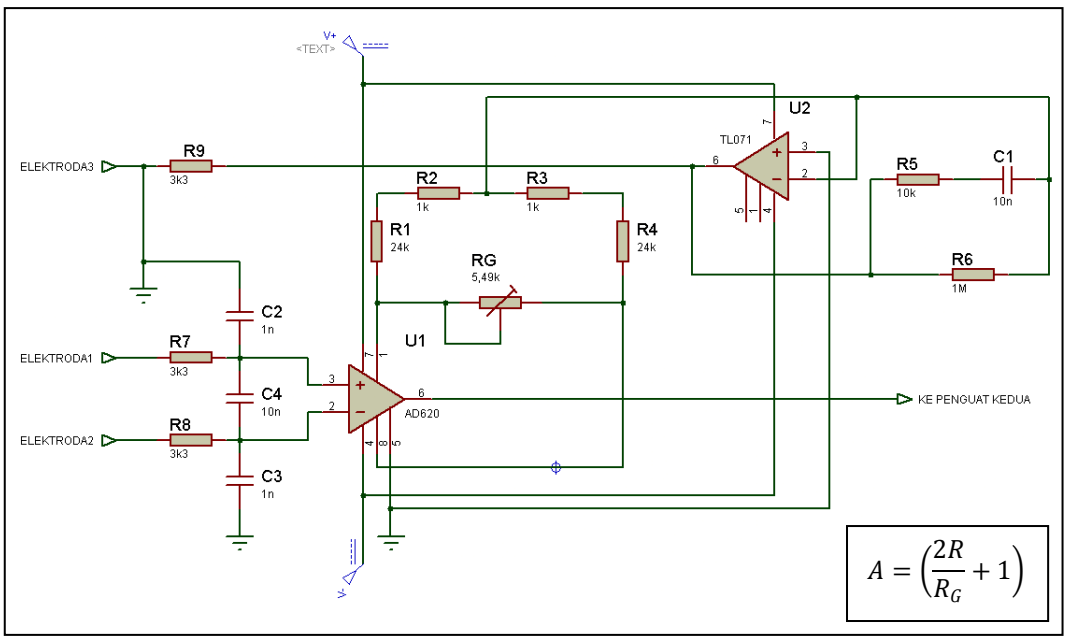
3.3.2 Perancangan Rangkaian

Ada beberapa rangkaian yang akan dirancang di dalam sistem, yaitu instrumentasi EMG yang terdiri atas rangkaian penguat awal (penguat instrumentasi), rangkaian penguat kedua (penguat operasional), rangkaian *filter*

rangkaian *clamper* dan rangkaian penguat akhir (penguat operasional); rangkaian sistem minimum mikrokontroler, rangkaian penerima dan pengirim data serial dan rangkaian DC *converter*.

1) Rangkaian Penguat Awal (Penguat Instrumentasi) dengan Gain 10 Kali

Berdasarkan karakteristik *Electromyography* (EMG), sinyal otot yang terbaca oleh sensor elektroda memiliki amplitudo yang sangat kecil, yaitu berkisar antara 0-10 mV. Maka itu dibutuhkan rangkaian penguat untuk menguatkan amplitudo sinyal otot agar dapat terbaca oleh ADC pada mikrokontroler. Penguatan ini dilakukan secara bertingkat (*cascade*). Penguatan dibagi ke dalam tiga kali penguatan. Pada penguat awal menggunakan penguat instrumentasi. Penguat instrumentasi ini lebih banyak digunakan dalam instrumentasi biomedik daripada penguat operasional biasa. Untuk penguat awal, IC yang digunakan adalah tipe AD620. Pada **Gambar 3.5** berikut adalah rangkaian dasar dari penguat instrumentasi dengan menggunakan IC AD620.



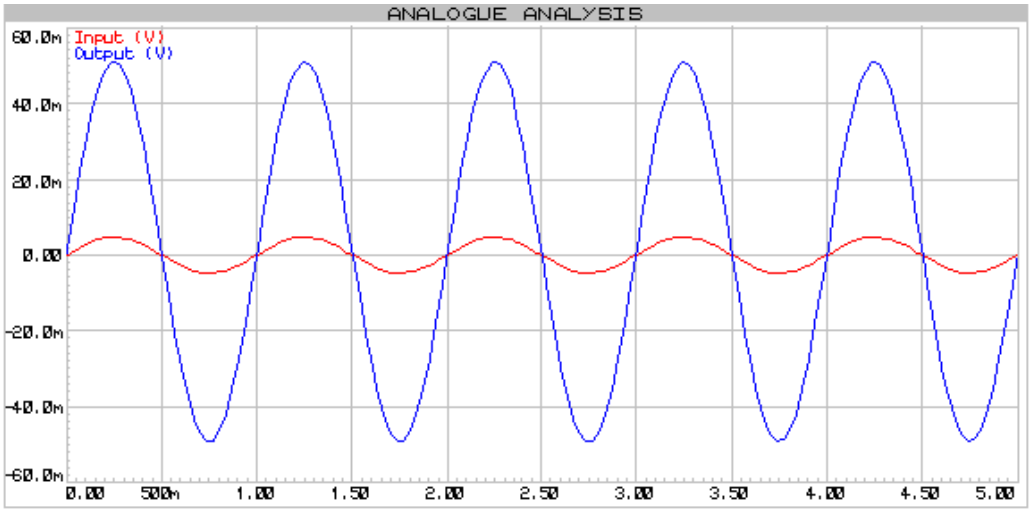
Gambar 3.5. Rangkaian Penguat Instrumentasi AD620

Untuk merancang penguat instrumentasi dengan penguatan (A) sebesar 10 kali, maka nilai R_G dapat dicari dengan perhitungan di bawah ini, dimana berdasarkan *datasheet*-nya, IC AD620 ini mempunyai tahanan dalam R sebesar 24,7 KΩ.

A=10x

$$R_G = \frac{2R}{A-1} = \frac{2 \text{ x } 24{,}7 \text{ K}\Omega}{10-1} = \frac{49{,}4 \text{ K}\Omega}{9} \approx 5{,}48 \text{ K}\Omega$$

Kemudian Pada **Gambar 3.6** berikut adalah realisasi hasil penguat instrumentasi yang sudah disimulasikan dengan *software* ISIS dari Proteus.

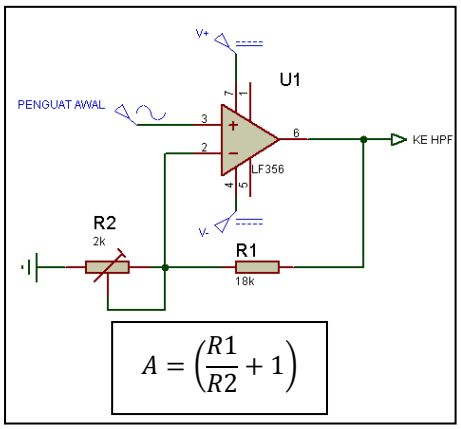


Gambar 3.6. Hasil Realisasi Penguat Instrumentasi dengan *Gain* 10 Kali

Pada **Gambar 3.6** di atas sinyal berwarna merah adalah sinyal asli (*input*) dengan amplitudo 10 mVpp, dan sinyal berwarna biru adalah sinyal yang dikuatkan (*output*) dengan rangkaian penguat instrumentasi dengan hasil amplitudo 100 mVpp. Hasil realisasi penguat instrumentasi tersebut sesuai dengan yang diharapkan, yaitu memiliki A=100 mVpp/10 mVpp=10.

2) Rangkaian Penguat Kedua (Penguat Operasional) dengan Gain 10 Kali

Pada penguat kedua ini dirancang dengan penguatan (*gain*) sebesar 10 kali. Penguat ini menggunakan jenis penguat operasional dengan metode *non-inverting*. Pada **Gambar 3.7** berikut adalah rangkaian penguat kedua (penguat operasional *non-inverting*) yang telah dirancang.

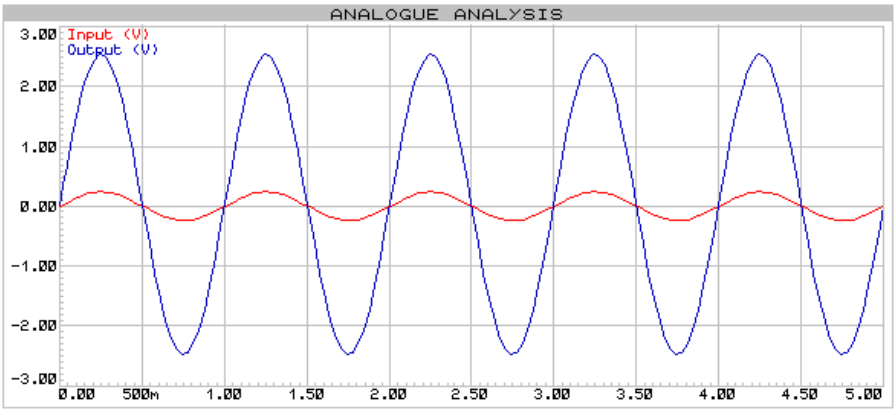


Gambar 3.7. Rangkaian Penguat Kedua (Penguat Operasional *Non-Inverting*)

Jika A=10x dan R2= 2 KΩ, maka nilai R1 dapat dicari sebagai berikut.

$$R1 = \mathrm{R2}.(\mathrm{A} - 1) = 2\ \mathrm{K\Omega}.(10 - 1) = 2\ \mathrm{K\Omega}.9 \approx 18\ \mathrm{K\Omega}$$

Kemudian pada **Gambar 3.8** adalah hasil realisasi penguat kedua (penguat operasional *non-inverting*).



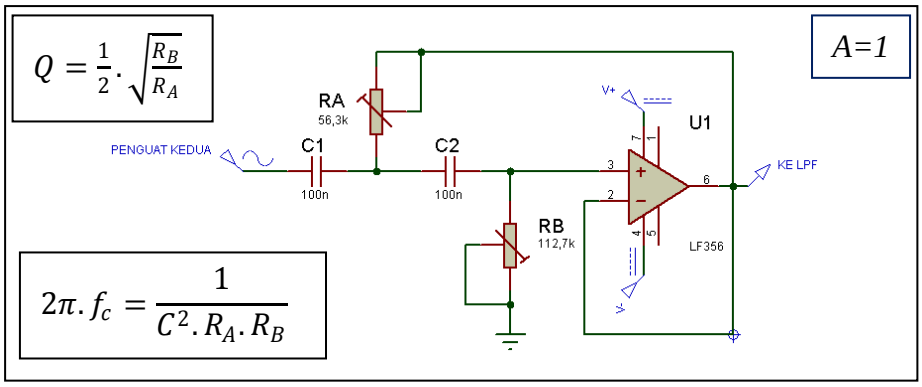
Gambar 3.8. Hasil Realisasi Penguat Kedua (Penguat Operasional *Non-Inverting*) dengan *Gain* 10 Kali

Pada **Gambar 3.8** di atas sinyal berwarna merah adalah sinyal asli (*input*) dengan amplitudo 500 mVpp, dan sinyal berwarna biru adalah sinyal yang mengalami penguatan (*output*) dengan amplitudo 5 Vpp. Hasil realisasi ini sesuai yang diharapkan, yaitu mengalami penguatan sebesar A=5 Vpp/500 mVpp=10.

3) Rangkaian High Pass Filter (HPF) 20 Hz

Berdasarkan karakteristiknya sinyal EMG memiliki *range* data pada frekuensi 20-500 Hz, maka itu diperlukan proses *filter* agar sinyal data yang dapat dilewatkan berada dikisaran 20-500 Hz. Salah satu *filter* yang akan digunakan adalah *high pass filter* (HPF) dengan frekuensi *cut-off* 20 Hz. Pada *filter* ini menggunakan respon *Butterworth* karena memiliki respon frekuensi berbentuk *flat* atau datar. Dan desain *filter* yang digunakan adalah *sallen-key* orde 2, karena memiliki desain yang sederhana dan memiliki *gain* yang positif karena menggunakan metode *non-inverting amplifier*.

IC yang digunakan untuk proses *filter* ini adalah jenis op-amp LF356 yang memiliki *bandwidth* yang besar, yaitu 5 MHz. IC ini cukup digunakan untuk proses *filter* dengan *bandwidth* yang dihasilkan dari sinyal EMG, yaitu sekitar 500-20= 480 Hz. Pada **Gambar 3.9** berikut adalah rangkaian *high pass filter* (HPF).



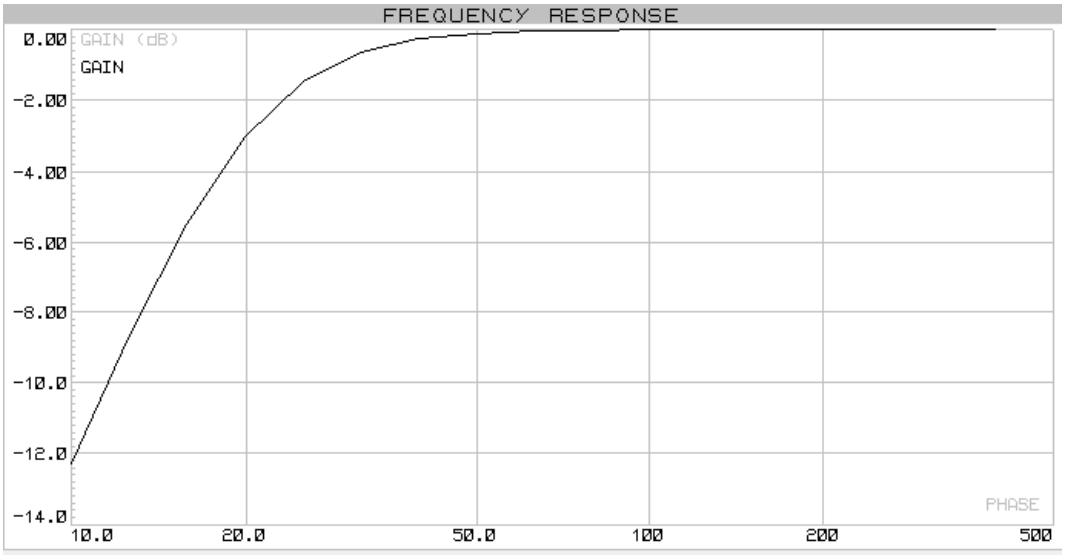
Gambar 3.9. Rangkaian *High Pass Filter* (HPF) *Sallen Key* Orde 2

Dari **Gambar 3.9** di atas jika nilai C_1=C_2=C=0,1 uF, Q=0,707 dan f_c=20 Hz, maka R_A dan R_B dapat dicari sebagai berikut.

$$R_A = \frac{1}{2.Q.2\pi.f_c.C} = \frac{1}{2.0{,}707.2\pi.20.0{,}1.10^{-6}} = \frac{1}{17{,}76.10^{-6}} \approx 56{,}3\ K\Omega$$

$$R_B = \frac{2.Q}{2\pi.f_c.C} = \frac{2.0{,}707}{2\pi.20.0,1.10^{-6}} = \frac{1{,}414}{12{,}56.10^{-6}} \approx 112{,}6\ K\Omega$$

Kemudian pada **Gambar 3.10** adalah hasil realisasi rangkaian HPF (*high pass filter*) dengan frekuensi *cut-off* 20 Hz.

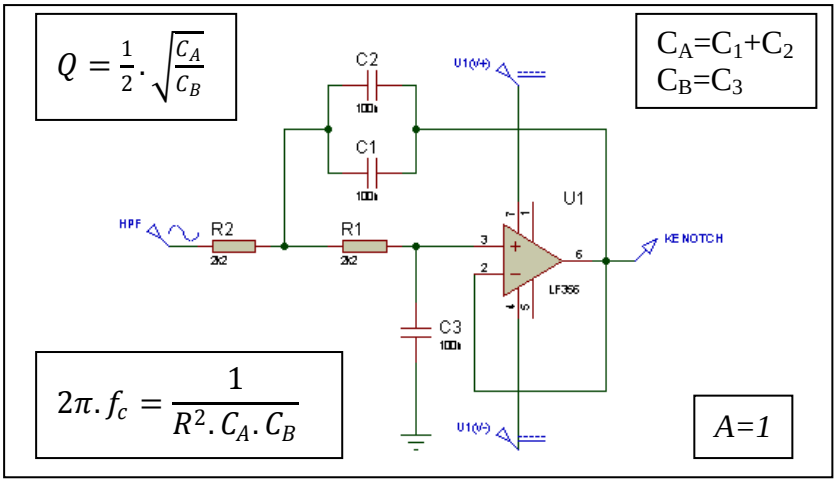


Gambar 3.10. Hasil Realisasi Rangkaian HPF dengan Frekuensi *Cut-off* 20 Hz

Pada **Gambar 3.10** di atas terlihat bahwa pada saat frekuensi lebih dari 20 Hz, maka frekuensi akan diloloskan, dan frekuensi kurang dari 20 Hz akan diredam. Hasil realisasi ini sesuai yang diharapkan.

4) Rangkaian LowPass Filter (LPF) 500 Hz

Filter ini bertujuan untuk meredam frekuensi di atas frekuensi *cut-off*-nya, yaitu 500 Hz. Pada **Gambar 3.11** berikut adalah rangkaian LPF yang digunakan dengan respon *Butterworth* dan desain *Sallen-key* orde 2.



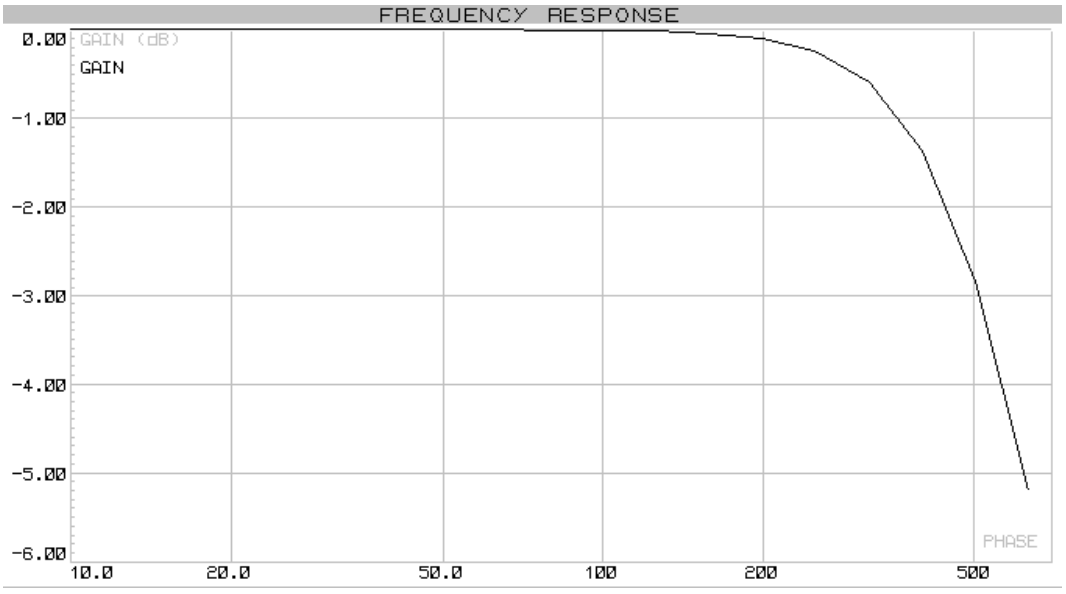
Gambar 3.11. Rangkaian *Low Pass Filter* (LPF) *Sallen Key* Orde 2

Dari **Gambar 3.11** di atas jika nilai R_1=R_2=R, Q=0,707, C_B=0,1 uF dan f_c=500 Hz, maka C_A dan R dapat dicari sebagai berikut.

$$C_A = 4.C_2.Q^2 = 4.0{,}1.10^{-6}.(0{,}707)^2 \approx 0{,}2.10^{-6}\ F = 0{,}2\ uF$$

$$R = \frac{1}{2\pi.f_c.\sqrt{C1.C2}} = \frac{1}{2\pi.500.\sqrt{0{,}1.10^{-6}.0{,}2.10^{-6}}} = \frac{1}{1000\pi.0{,}14.10^{-6}} \approx 2{,}25\ K\Omega$$

Kemudian pada **Gambar 3.12** adalah hasil realisasi dari rangkaian LPF (*low pass filter*) dengan frekuensi *cut-off* 500 Hz.

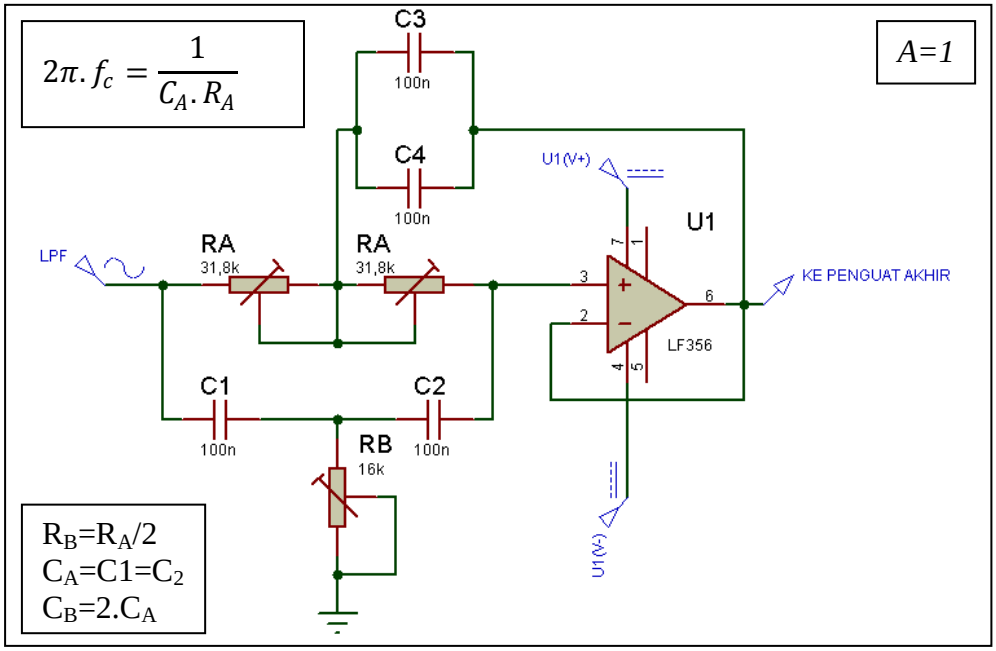


Gambar 3.12. Hasil Realisasi rangkaian LPF dengan Frekuensi *Cut-off* 500 Hz

Pada **Gambar 3.12** di atas terlihat bahwa pada saat frekuensi lebih dari 500 Hz, maka frekuensi akan diredam, dan frekuensi kurang dari 500 Hz akan diloloskan. Hasil realisasi ini sesuai yang diharapkan.

5) Rangkaian Notch Filter 50 Hz

Rangkaian *Notch filter* ini berfungsi untuk meredam sinyal pada frekuensi tertentu. Pada perancangan ini sinyal yang akan diredam berada pada frekuensi 50 Hz. Karena frekuensi ini berasal dari jala-jala listrik yang akan menjadi *noise* atau sinyal pengganggu. Pada **Gambar 3.13** berikut adalah rangkaian *Notch filter* yang dirancang dengan respon *Butterworth* dan desain *Sallen-key* orde 2.



Gambar 3.13. Rangkaian *Notch Filter Sallen Key* Orde 2

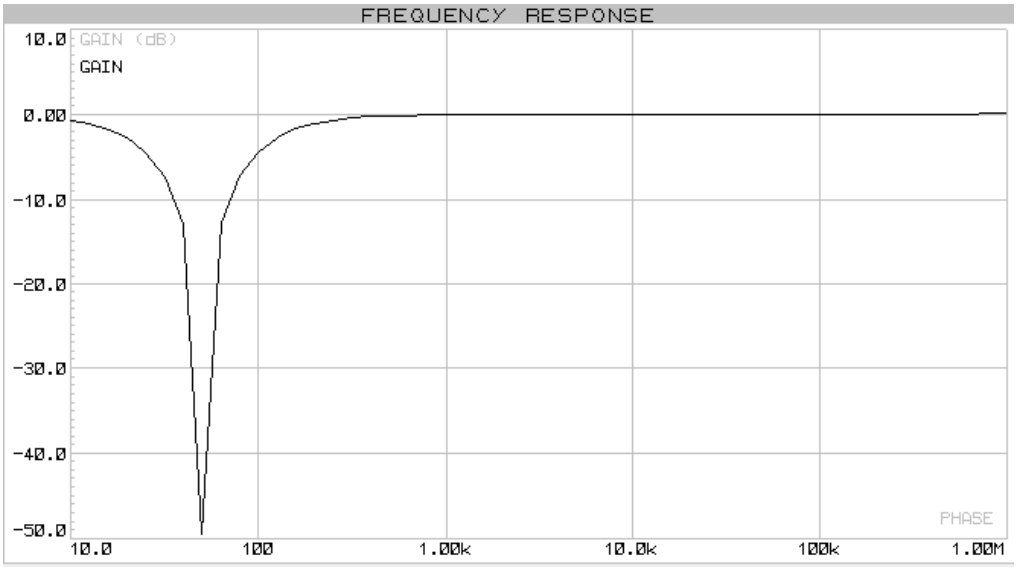
Jika pada rangkaian di atas parameter C_A=0,1 uF dan f_c=50 Hz, maka nilai R_A dapat ditentukan sebagai berikut.

$$R_A = \frac{1}{C . 2\pi . f_c} = \frac{1}{0.1 . 10^{-6} . 2\pi . 50} \approx 31{,}8\ K\Omega$$

$R_B=R_A/2=15{,}9\ K\Omega$

$C_B=2.C_A=0,2$ uF

Kemudian pada **Gambar 3.14** berikut adalah hasil realisasi rangkaian *notch filter* dengan frekuensi *cut-off* 50 Hz.

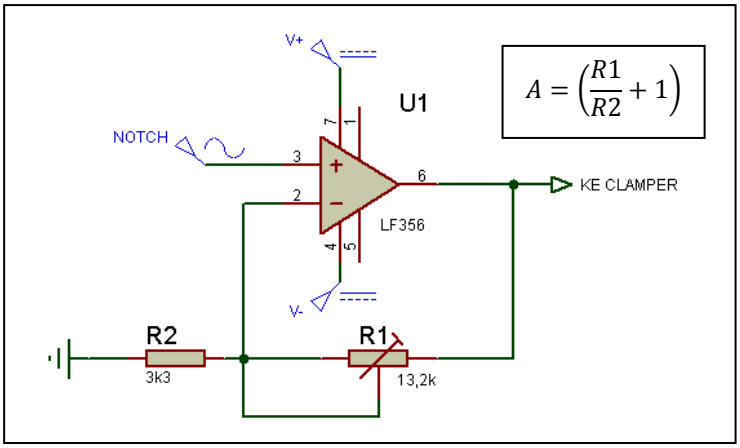


Gambar 3.14. Hasil Realisasi Rangkaian *Notch Filter* dengan Frekuensi *Cut-off* 50 Hz

Pada **Gambar 3.14** di atas terlihat bahwa pada frekuensi 50 Hz, maka frekuensi tersebut diredam, dan frekuensi selain 50 Hz akan diloloskan. Hasil realisasi ini sesuai dengan yang diharapkan.

6) Rangkaian Penguat Operasional Akhir dengan Gain 5 Kali

Rangkaian penguat operasional akhir ini adalah rangkaian penguat sinyal yang terakhir dirancang. Penguat operasional akhir ini bertujuan agar sinyal yang dikuatkan dapat terbaca oleh ADC (*analog to digital converter*) pada mikrokontroler. Rangkaian ini dirancang dengan *gain* sebesar 5 kali. Penguat ini menggunakan metode *non-inverting.* Pada **Gambar 3.15** berikut adalah rangkaian penguat operasional akhir yang telah dirancang.

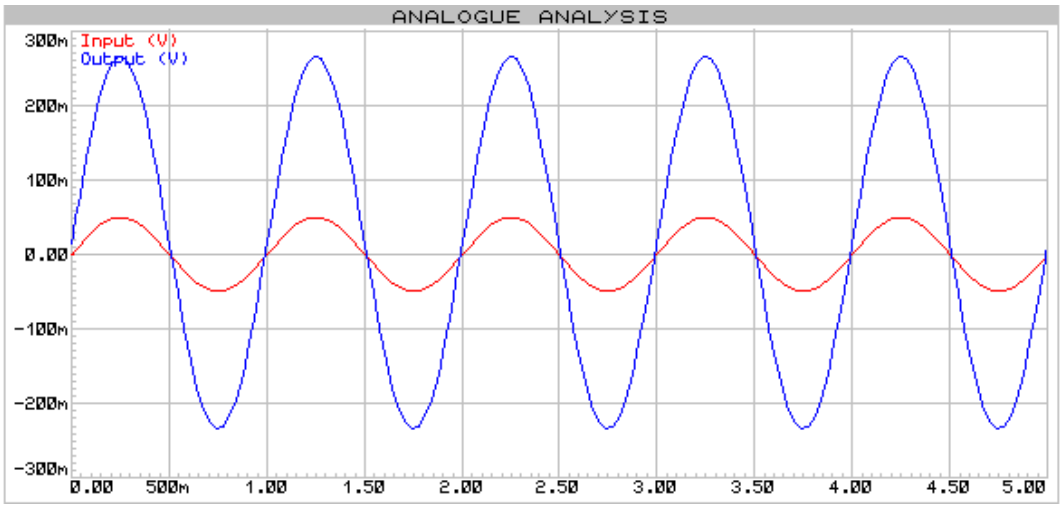


Gambar 3.15. Rangkaian Penguat Operasional Akhir *Non-Inverting*

A=5x dan jika R2= 3,3 KΩ

$R1 = \mathrm{R2}.(\mathrm{A} - 1) = 3{,}3\ \mathrm{K\Omega}.(5 - 1) = 3{,}3\ \mathrm{K\Omega}.4 \approx 13{,}2\ \mathrm{K\Omega}$

Kemudian pada **Gambar 3.16** berikut adalah hasil realisasi dari rangkaian penguat operasional akhir yang dirancang dengan penguatan sebesar 5 kali.



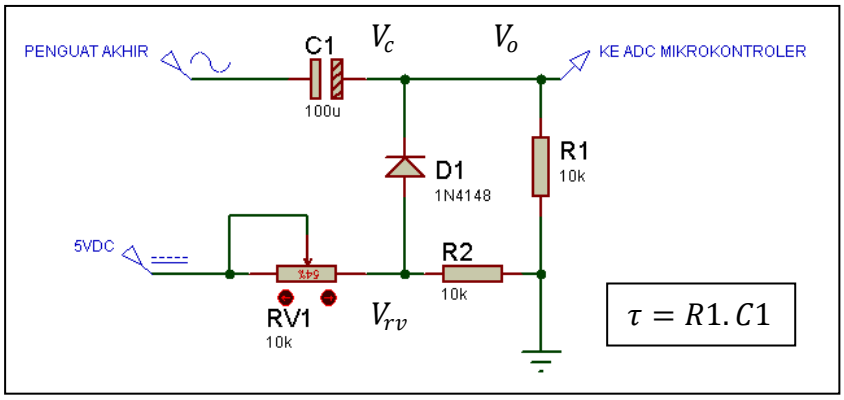
Gambar 3.16. Hasil Realisasi Penguat Operasional Akhir *Non-Inverting* dengan *Gain* 5 Kali

Pada **Gambar 3.16** di atas sinyal berwarna merah adalah sinyal asli (*input*) dengan amplitudo 100 mVpp, dan sinyal berwarna biru adalah sinyal yang sudah

dikuatkan (*output*) dengan penguat operasional awal *non-inverting* yang memiliki amplitudo 500 mVpp. Hasil realisasi sesuai yang diharapkan, yaitu memiliki *gain* A=500 mVpp/100 mVpp=5.

7) Rangkaian Clamper up to 5 V

Pada tahap akhir karena sinyal data yang dihasilkan dari proses sebelumnya masih memiliki amplitudo yang negatif, maka digunakan rangkaian *clamper* positif yang berfungsi untuk menggeser nilai tegangan agar berada di atas nol atau positif tanpa mengubah amplitudo dari sinyal data yang didapat. Ini ditujukan agar sinyal data yang didapat tidak terpotong. Pada **Gambar 3.17** adalah rangkaian *clamper* positif yang dirancang.



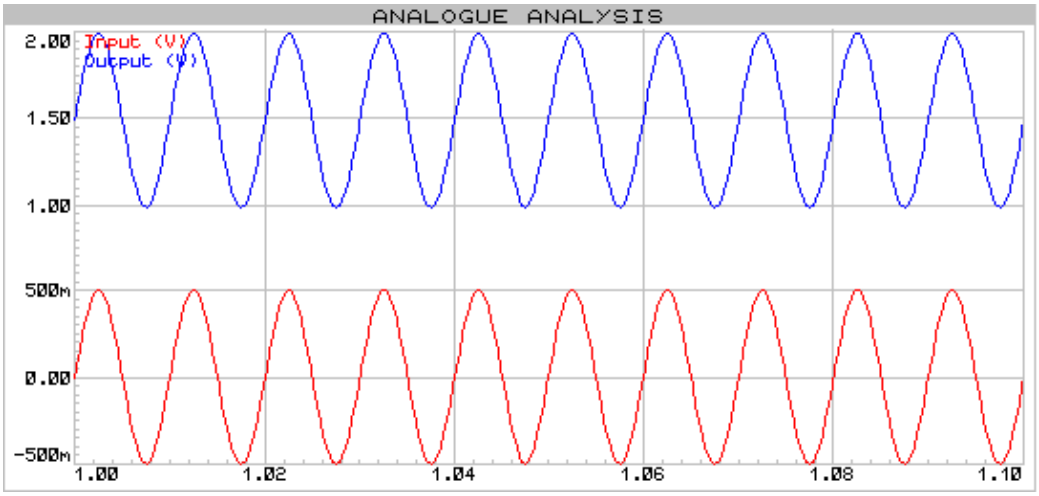
Gambar 3.17. Rangkaian *Clamper* Positif

Pada rangkaian *clamper* ini pada saat dioda (D1) dibias mundur atau diberi tegangan negatif, maka dioda tersebut akan mengalami *open circuit* dan kapasitor akan melepas muatan tegangan yang ada, sehingga nilai tegangan *output* akan sama dengan $V_o = V_{rv} + (-V_c)$. Dan pada saat dioda dibias maju atau diberi tegangan positif, maka dioda akan mengalami *short circuit* dan kapasitor akan mengisi muatan tegangan V_c, sehingga $V_o = V_{rv} + V_c$. Nilai resistor (R1) dan kapasitor (C1) ditentukan sehingga konstanta waktu RC yang tepat agar tidak

terjadi pengosongan muatan yang terlalu cepat atau terlalu lambat saat dioda tidak menghantar. Jika nilai R1=10 KΩ dan C1=100 uF, maka,

$\tau = R1.C1 = 10\ K\Omega.100\ \text{uF} = 1\ s$

Untuk komponen potensiometer (RV1) dan resistor (R2) dirangkai seri sebagai bentuk rangkaian pembagi tegangan. Rangkaian pembagi tegangan ini berfungsi untuk mengatur seberapa jauh *based line* sinyal yang akan digeser. Maksimal penggeseran *based line* sinyal adalah sebesar nilai VCC yang digunakan. Pada rangkaian *clamper* yang dirancang ini menggunakan VCC 5V. Pada **Gambar 3.18** berikut adalah hasil realisasi dari rangkaian *clamper* positif yang dirancang.



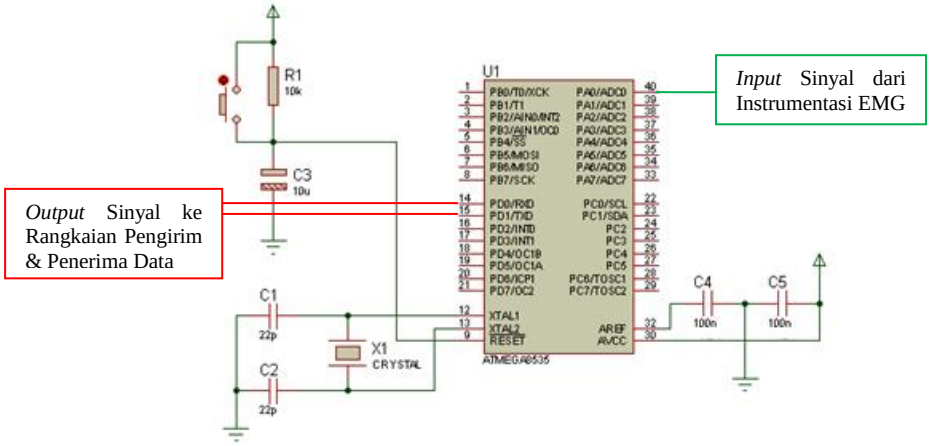
Gambar 3.18. Hasil Realisasi Rangkaian *Clamper* Positif

Pada **Gambar 3.18** di atas sinyal berwarna merah adalah sinyal asli dengan *based line* pada 0 V dan sinyal berwarna biru adalah sinyal yang sudah digeser *based line*-nya sejauh 1,5 V. Hasil realisasi ini sesuai yang diharapkan.

8) Rangkaian Sistem Minimum Mikrokontroler

Rangkaian sistem minimum merupakan rangkaian yang dirancang

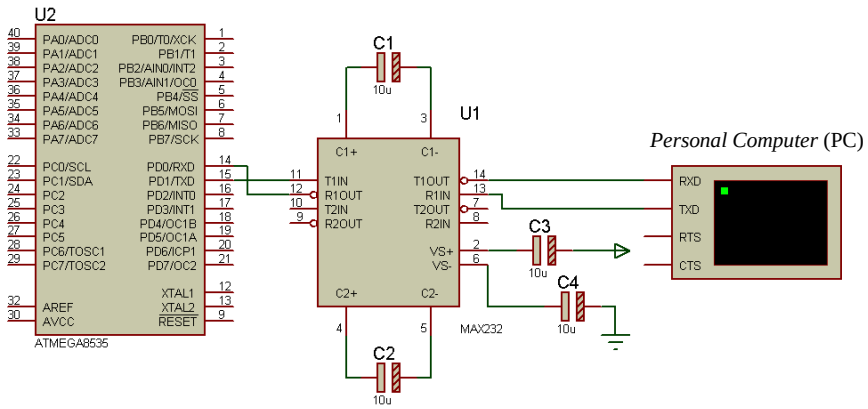
menggunakan komponen yang sedikit untuk mendukung kerja mikrokontroler sesuai yang diinginkan. Pada **Gambar 3.19** berikut adalah rangkaian sistem minimum mikrokontroler yang digunakan.



Gambar 3.19. Rangkaian Sistem Minimum Mikrokontroler

Kristal eksternal yang digunakan pada rangkaian sistem minimum di atas adalah 12 MHz, walaupun mikrokontroler sudah memiliki kristal internal. Kristal ini berfungsi sebagai jantung utama dari sebuah mikrokontroler. Mikrokontroler yang digunakan adalah ATMega 8535 yang mempunyai resolusi ADC sebesar 10 bit.

9) Rangkaian Pengirim dan Penerima Data

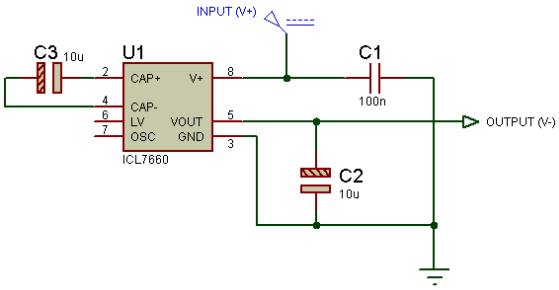


Gambar 3.20. Rangkaian Pengirim dan Penerima Data Serial

Pada **Gambar 3.20** di atas.adalah rangkaian pengirim dan penerima data serial yang digunakan. Rangkaian ini berfungsi untuk mengirim data ke mikrokontroler ataupun menerima data serial yang dikirim dari mikrokontroler. Rangkaian ini mengirim data per-bit dalam waktu tertentu secara terus menerus. Pada rangkaian di atas menggunakan komponen MAX232 sebagai antarmuka komunikasi serial RS-232. IC ini menggunakan *supply* tegangan yang sama dengan mikrokontroler yang digunakan, yaitu 5 V.

10) Rangkaian DC Converter

Rangkaian DC *converter* adalah rangkaian pembalik tegangan positif untuk menghasilkan tegangan yang negatif. Rangkaian ini dibutuhkan sebagai *supply*/catu daya simetris DC untuk rangkaian penguat instrumentasi, penguat operasional awal dan akhir serta rangkaian *filter* yang menggunakan IC AD620 dan LF356. IC tersebut membutuhkan *supply* simetris untuk beroperasi. Rangkaian ini menggunakan IC ICL7660 sebagai komponen utamanya. Pada **Gambar 3.21** berikut adalah rangkaian DC *converter* yang dirancang.



Gambar 3.21. Rangkaian DC *Converter*

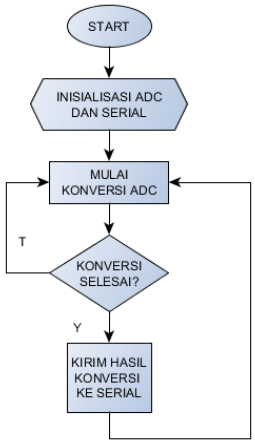
3.4 Perancangan *Software*

Pada perancangan *software* ini akan menjelaskan semua fungsi dari aplikasi yang akan dibuat dan digunakan. *Software* yang akan digunakan dalam

perancangan *software* adalah LabVIEW 8.5. Dalam perancangan *software* ini terbagi menjadi beberapa bagian, yaitu perancangan perangkat lunak pada mikrokontroler 1, akuisisi data/pengambilan data, *filter* digital, ekstraksi fitur, klasifikasi data menggunakan ANN dan perancangan perangkat lunak pada mikrokontroler 2.

3.4.1 Perancangan Perangkat Lunak Mikrokontroler 1 (ADC)

Perancangan perangkat lunak (*firmware*) mikrokontroler 1 ini ditujukan untuk proses ADC (*analog to digital converter*) dari sinyal EMG yang terbaca dan mengirimnya secara serial ke PC/*personal computer* untuk proses akuisisi data. Bahasa program yang digunakan adalah *assembly*. Pada **Gambar 3.22** berikut adalah diagram alir *firmware* pada mikrokontroler 1.



Gambar 3.22. Diagram Alir *Firmware* Mikrokontroler 1

Pada mikrokontroler ATMega8535 sebelum melakukan proses ADC perlu dilakukan inisialisasi ADC dan *baudrate* sebagai berikut.

1. Pengaturan register ADMUX, yaitu register 8-bit yang berfungsi untuk menetapkan tegangan referensi ADC, format data keluaran dan *channel* ADC yang akan digunakan. Tegangan referensi yang digunakan adalah Vref

eksternal (REFS1=0; REFS0=1) dengan besar 5 V, pengaturan rata kiri hasil ADC (ADLAR:1) dan hanya satu *channel* yang digunakan untuk melakukan proses ADC (MUX4:0=00000). Berikut adalah konfigurasinya.

Tabel 3.5. Konfigurasi Register ADMUX

REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0
0	1	1	0	0	0	0	0

2. Pengaturan register ADCSRA (ADC *Control and Status Register* A), yaitu register 8-bit yang berfungsi melakukan manajemen sinyal kontrol, status dari ADC dan kecepatan frekuensi *clock* ADC (ADPS2:0=111). Berikut adalah konfigurasinya.

Tabel 3.6. Konfigurasi Register ADCSRA

ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0
1	0	0	0	0	1	1	1

Untuk mengatur aktivasi ADC maka bit ADEN (ADC *Enable*) di-set *high* dan untuk memulai konversi ADC maka bit ADSC (ADC *Start Convertion*) di-set *high*. Untuk menentukan kecepatan frekuensi *clock* ADC, maka perlu dilakukan pengaturan bit ADPS2:0. Berikut adalah konfigurasi kecepatan *clock* ADC.

Tabel 3.7. Konfigurasi Kecepatan *Clock* ADC

ADPS2:0	000	001	010	011	100	101	110	111
Clock ADC	-	$f_{osc}/2$	$f_{osc}/4$	$f_{osc}/8$	$f_{osc}/16$	$f_{osc}/32$	$f_{osc}/64$	$f_{osc}/128$

f_{osc} adalah frekuensi kristal yang digunakan (12 MHz). Karena ATMega8535 hanya dapat merespon frekuensi tidak lebih dari 200 KHz, maka bit ADPS2:0 yang dipilih adalah 111, sehingga frekuensi *clock* ADC

adalah f_{osc}/128=93,75KHz.

3. Pengaturan frekuensi *sampling*. Berdasarkan karakteristik dari EMG yang memiliki rentang frekuensi dari 20 Hz-500 Hz, maka menurut teori Nyquist frekuensi *sampling* minimal adalah dua kali dari frekuensi maksimum, yaitu 1000 Hz. Frekuensi *sampling* yang akan digunakan adalah sebesar 2000 Hz. Untuk mendapatkan frekuensi *sampling* sebesar 2000 Hz, hasil konversi ADC akan dibaca setiap waktu yang ditentukan, yaitu 1/2000 Hz = 0,5 mS. Jadi, dalam satu detik hasil konversi ADC yang dapat dibaca adalah sebesar 1/0,5 mS=2000 sampel.
4. Pengaturan *baudrate*, yaitu parameter untuk mengatur kecepatan data ADC yang akan dikirim secara serial. Pengaturan ini bertujuan untuk proses akuisisi data. Berikut adalah konfigurasinya.
    a. *Baudrate* yang akan digunakan adalah sebesar 57,6 Kbps dan data serial yang dikirim adalah 10-bit (8-bit data serial, 1-bit *start* dan 1-bit *stop*). Jadi, data yang dapat dikirim setiap detiknya adalah 57,6 Kbps/10-bit=5,76 KHz.
    b. Berdasarkan teori Nyquist, frekuensi maksimum yang bisa dikirim adalah 5,76 KHz/2=2,88 KHz. Karena frekuensi *sampling* yang digunakan adalah 2 KHz dan kurang dari frekuensi maksimum yang bisa dikirim, maka nilai *baudrate* yang dipilih bisa digunakan untuk proses akuisisi data.

3.4.2 Perancangan Program Aplikasi Komputer

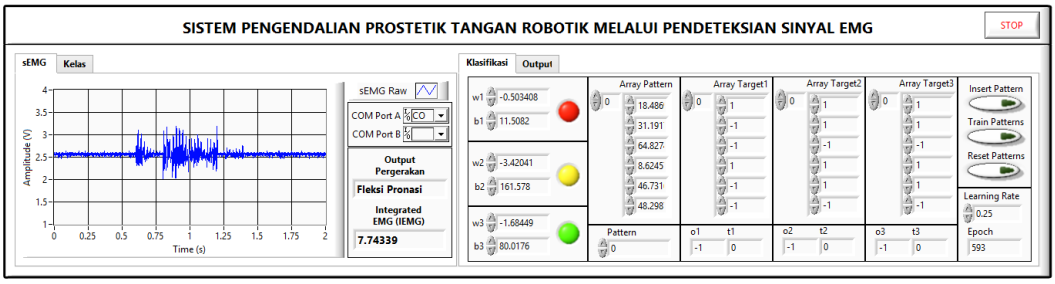
Perancangan program aplikasi komputer ini dibuat menggunakan *software* LabVIEW 8.5. *Software* ini adalah salah satu aplikasi yang dapat membuat

program menggunakan notasi grafik (menghubungkan node-node fungsi dimana data mengalir). Program ini dibuat untuk membaca *raw* data dari sinyal EMG yang telah dikirim secara serial oleh mikrokontroler 1, kemudian analisis dengan proses ekstraksi fitur pada domain waktu dan proses klasifikasi menggunakan *artificial neural network* (ANN). Pada **Gambar 3.23** berikut adalah blok diagram dari perancangan program aplikasi komputer yang dibuat.



Gambar 3.23. Blok Diagram Program Aplikasi Komputer

Gambar 3.24 berikut adalah tampilan utama dari program aplikasi komputer yang sudah dirancang.



Gambar 3.24. Tampilan Utama Program Aplikasi Komputer

3.4.2.1 Perancangan Filter Digital

Dalam pengolah sinyal EMG dibutuhkan beberapa *filter* untuk meredam sinyal yang tidak termasuk dalam karakteristik sinyal EMG, baik itu *filter* analog maupun *filter* digital. Karakteristik sinyal EMG berada pada frekuensi 20 Hz-500 Hz dan sudah di-*filter* pada rangkaian *filter* analog. Pada rangkaian *filter* analog, *noise* pada sinyal EMG dapat dikurangi, tetapi untuk pengolahan sinyal EMG yang sudah digitalisasi, pembacaan ADC juga berpengaruh pada sinyal EMG

sehingga ada kemungkinan *noise* masih ada dalam sinyal EMG yang didapatkan. Ini disebabkan karena pada dasarnya *filter* yang benar-benar ideal itu tidak ada. Oleh karena itu, akan dirancang sebuah *filter* secara digital untuk mengurangi *noise* yang kemungkinan masih lolos pada *filter* analog sebelumnya.

Berikut ini adalah perancangan *filter* IIR *band pass Butterworth* yang akan dirancang.

1. Spesifikasi *filter* yang akan dirancang memiliki frekuensi *pass band* f_{p1}=60 Hz dan f_{p2}=250 Hz, frekuensi *stop band* f_{s1}=10 Hz dan f_{s2}=300 Hz, frekuensi *sampling* ($f_{sampling}$) 700 Hz, *ripple pass band* maksimum (r_p) -3 dB dan atenuasi minimum *stop band* (r_s) -8 dB.
2. Melakukan *prewarp* frekuensi dari digital ke analog.

$$w_{p1} = \tan\left(\frac{\pi . f_{p1}}{f_{sampling}}\right) = 0{,}27\ rad/detik$$

$$w_{p2} = \tan\left(\frac{\pi . f_{p2}}{f_{sampling}}\right) = 2{,}04\ rad/detik$$

$$w_{s1} = \tan\left(\frac{\pi . f_{s1}}{f_{sampling}}\right) = 0{,}044\ rad/detik$$

$$w_{s2} = \tan\left(\frac{\pi . f_{s2}}{f_{sampling}}\right) = 4{,}47\ rad/detik$$

3. Men-transformasikan ke dalam *low pass prototype*.

$$w_{A1} = \frac{{w_{s1}}^2 - (w_{p1} . w_{p2})}{w_{s1}(w_{p1} - w_{p2})} = 7{,}05$$

$$w_{A2} = \frac{{w_{s2}}^2 - (w_{p1} . w_{p2})}{w_{s2}(w_{p1} - w_{p2})} = -2{,}456$$

4. Mencari nilai orde dari *low pass prototype* yang didapat.

$$w_A = \min(|w_{A1}|, |w_{A2}|) = 2{,}456$$

$$n \geq \frac{\log\left(\frac{\left(10^{0,1.|r_s|}-1\right)}{\left(10^{0,1.|r_p|}-1\right)}\right)}{2.\log(w_A)} = 1$$

5. Menentukan fungsi transfer H(s) berdasarkan table denominator polinomial *Butterworth* dengan nilai orde yang didapat, yaitu n=1.

$$H(s) = \frac{1}{B_1(s)} = \frac{1}{s+1}$$

$$s = \frac{a^2 + w_{p1}w_{p2}}{a\left(w_{p2} - w_{p1}\right)} = \frac{a^2 + 0{,}55}{1{,}77a}$$

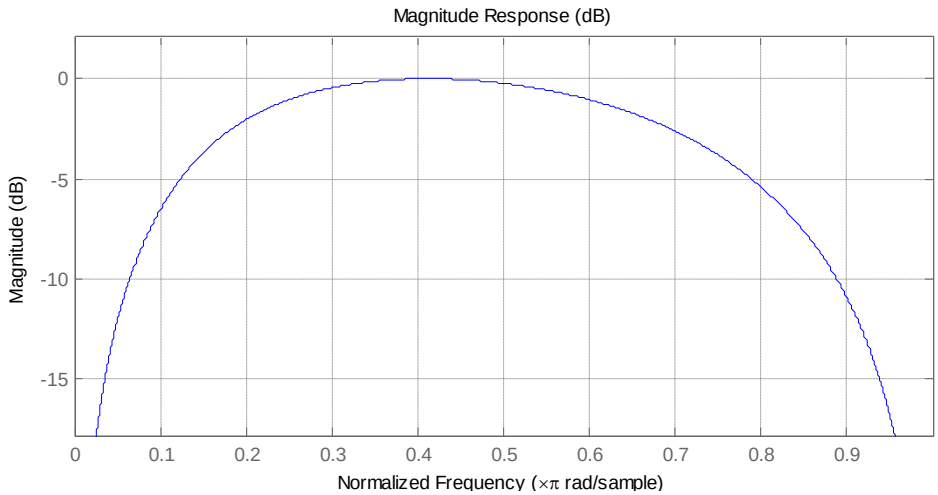
$$a = \frac{z-1}{z+1} \equiv \frac{1-z^{-1}}{1+z^{-1}}$$

$$H(z) = \frac{0{,}53 - 0{,}53z^{-2}}{1 - 0{,}27z^{-1} - 0{,}07z^{-2}}$$

6. Mengubah fungsi H(z) yang didapat ke dalam persamaan beda y(n).

$$y[n] = 0{,}53x[n] - 0{,}53x[n-2] + y[n] - 0{,}27y[n-1] - 0{,}07y[n-2]$$

Kemudian pada **Gambar 3.25** berikut adalah hasil realisasi *filter* yang telah dirancang.



Gambar 3.25. Hasil Realisasi Respon Frekuensi *Filter* IIR *Band Pass Butterworth* dalam frekuensi ternormalisasi

3.4.2.2 Perancangan Proses Ekstraksi Fitur Sinyal EMG

Sebelum dilakukan proses klasifikasi maka *raw* data yang telah di-*filter* perlu dilakukan proses ekstraksi fitur untuk menghasilkan nilai keluaran yang bisa dijadikan *input*/masukan untuk proses klasifikasi selanjutnya. Proses ini dilakukan dalam domain waktu. Dari banyak fitur yang bisa diekstraksi dari sinyal EMG. *Integrated* EMG (IEMG) akan dijadikan *input* dalam proses klasifikasi.

Integrated EMG (IEMG) dihitung berdasarkan penjumlahan dari nilai absolut dari amplitudo sinyal EMG. Pada umumnya, IEMG digunakan sebagai indeks untuk mendeteksi aktivitas otot sebagai pengontrol devais. IEMG dapat dirumuskan sebagai berikut.

$$IEMG = \sum_{n=1}^{N} |x_n| \qquad (3.2)$$

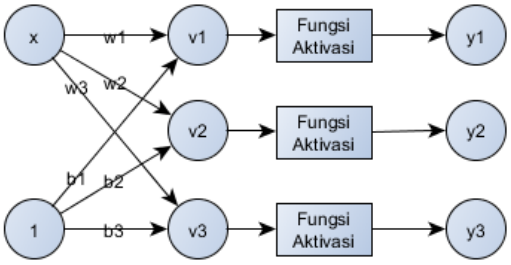
Dimana: N = jumlah sampel

n = nomor sampel

x_n = amplitudo sampel sinyal ke-n

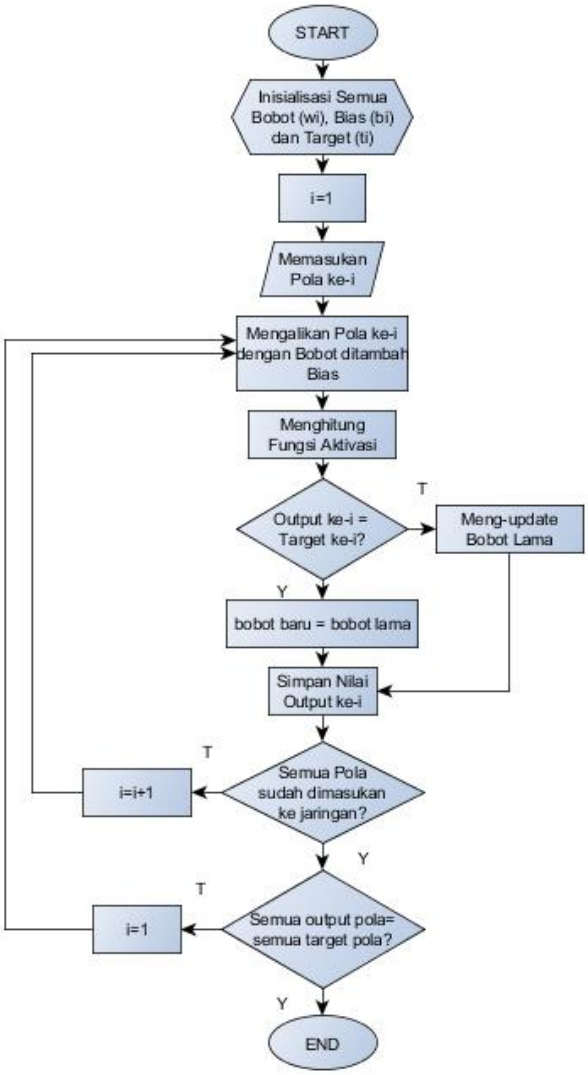
3.4.2.3 Perancangan Proses Klasifikasi dengan Artificial Neural Network (ANN)

Perancangan ANN ditujukan untuk meng-klasifikasikan nilai ekstraksi fitur EMG yang didapat sebelumnya untuk menentukan gerakan tangan yang sedang dilakukan. Pada **Gambar 3.26** berikut adalah model jaringan *perceptron* yang dirancang.



Gambar 3.26. Model Jaringan *Perceptron*

Kemudian pada **Gambar 3.27** berikut adalah diagram alir/*flowchart* yang digunakan dalam proses pembelajaran dalam ANN.



Gambar 3.27. Diagram Alir pada Jaringan *Perceptron* yang dirancang

Dari **Gambar 3.27** dapat dijelaskan secara rinci sebagai berikut.

1. Banyaknya *input* adalah 1 (satu) yang diambil dari IEMG (x) pada proses ekstraksi fitur sebelumnya. Banyak *output* target yang diinginkan adalah 3 (tiga), yaitu gerakan ekstensi supinasi ($[t_1\ t_2\ t_3]=[1\ 1\ 1]$), gerakan fleksi supinasi ($[t_1\ t_2\ t_3]=[-1\ 1\ 1]$) dan gerakan fleksi pronasi ($[t_1\ t_2\ t_3]=[-1\ -1\ 1]$). Setiap *input* gerakan tangan diambil dua kali pengambilan data pola,

sehingga pola yang akan di lakukan pembelajaran ada sebanyak 6 (enam) pola. Pasangan pola ditunjukan pada **Tabel 3.8** berikut.

Tabel 3.8. Pasangan *Input* dan *Output* Target

Pola	**x**	**t_1**	**t_2**	**t_3**
1	IEMG Ekstensi Supinasi 1	1	1	1
2	IEMG Fleksi Supinasi 1	-1	1	1
3	IEMG Fleksi Pronasi 1	-1	-1	1
4	IEMG Ekstensi Supinasi 2	1	1	1
5	IEMG Fleksi Supinasi 2	-1	1	1
6	IEMG Fleksi Pronasi 2	-1	-1	1

2. Bobot dan bias pada awalnya diberikan nilai acak $-0,5 \leq w_i \leq 0,5$ dan $-0,5 \leq b_i \leq 0,5$ dengan laju pembelajaran $\alpha=0,25$.
3. Untuk proses pembelajaran pada jaringan dilakukan proses dengan algoritma sebagai berikut.
   a. Memasukan nilai pola pertama x, t_1, t_2, t_3 ke dalam jaringan.
   b. Menghitung nilai v_1, v_2 dan v_3 dengan rumus sebagai berikut.

$$v_1 = xw_1 + b_1 \qquad (3.3)$$

$$v_2 = xw_2 + b_2 \qquad (3.4)$$

$$v_3 = xw_3 + b_3 \qquad (3.5)$$

   c. Setiap nilai v_1, v_2 dan v_3 dihitung nilai aktivasinya untuk mendapatkan nilai *output* y_1, y_2 dan y_3 dengan rumus 3.6. Fungsi aktivasi yang digunakan adalah fungsi aktivasi *Piecewise-linier*. Nilai *threshold* yang digunakan adalah 1.

$$y_i = \begin{cases} 1 & jika\ v_i \geq 1 \\ 0 & jika -1 < v_i < 1 \\ -1 & jika\ v_i \leq 1 \end{cases} \qquad (3.6)$$

   d. Membandingkan nilai *output* y_1, y_2 dan y_3 dengan target t_1, t_2 dan t_3.
      - $y_1 \neq t_1$, maka perbaiki bobot dan bias w_1 dan b_1 dan jika $y_1 = t_1$, maka

tidak ada perbaikan bobot atau bobot baru=bobot lama dan bias baru=bias lama;

- $y_2 \neq t_2$, maka perbaiki bobot dan bias w_2 dan b_2 dan jika $y_2 = t_2$, maka tidak ada perbaikan bobot atau bobot baru=bobot lama dan bias baru=bias lama;
- $y_3 \neq t_3$, maka perbaiki bobot dan bias w_3 dan b_3 dan jika $y_3 = t_3$, maka tidak ada perbaikan bobot atau bobot baru=bobot lama dan bias baru=bias lama.

Untuk perbaikan bobot dan bias menggunakan rumus sebagai berikut.

$w_{1\ baru} = w_{1\ lama} + \Delta w$ dengan $\Delta w = \alpha . t_1 . x$ (3.7)

$w_{2\ baru} = w_{2\ lama} + \Delta w$ dengan $\Delta w = \alpha . t_1 . x$ (3.8)

$w_{3\ baru} = w_{3\ lama} + \Delta w$ dengan $\Delta w = \alpha . t_3 . x$ (3.9)

$b_{1\ baru} = b_{1\ lama} + \Delta b$ dengan $\Delta b = \alpha . t_1$ (3.10)

$b_{2\ baru} = b_{2\ lama} + \Delta b$ dengan $\Delta b = \alpha . t_2$ (3.11)

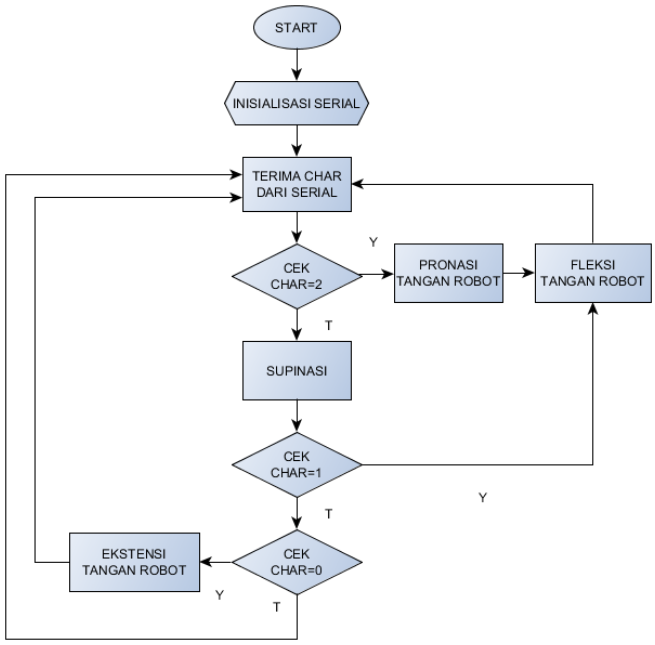
$b_{3\ baru} = b_{3\ lama} + \Delta b$ dengan $\Delta b = \alpha . t_3$ (3.12)

e. Memasukan pola x, t_1, t_2 dan t_3 selanjutnya dan ulangi langkah a-e. Untuk semua pola pertama sampai terakhir dihitung sebagai satu epok (iterasi). Proses akan berhenti setelah semua nilai y=t atau semua nilai *output* dari setiap pola bernilai sama dengan nilai target dari setiap pola.

3.4.3 Perancangan Perangkat Lunak Mikrokontroler 2 (Kontrol)

Perancangan perangkat lunak (*firmware*) pada mikrokontroler 2 ditujukan untuk mengatur gerakan tangan robot. Motor servo digunakan sebagai aktuator penggerak tangan robot yang sudut putarannya diatur dengan teknik PWM.

Mikrokontroler yang digunakan sendiri adalah ATMega8535 dengan bahasa program *assembly*. Pada **Gambar 3.28** berikut adalah diagram alir dari *firmware* pada mikrokontroler 2.



Gambar 3.28. Diagram Alir Program Mikrokontroler 2

Fleksi dan ekstensi tangan robot digerakan dengan motor servo yang dikontrol dengan teknik PWM. Untuk mengatur PWM pada mikrokontroler ATMega8535 ini maka perlu dilakukan pengaturan awal sebagai berikut.

1. Pengaturan nilai *timer* yang diinginkan pada register TCNT1. Register TCNT1 ini adalah register 16-bit yang dibagi menjadi 2 register 8-bit, yaitu TCNT1H dan TCNT1L. Berikut adalah rumus yang digunakan untuk mendapatkan nilai TCNT1.

$$TCNT1 = 65536 - (\frac{T x f_{clk}}{N}) \qquad (3.13)$$

Dimana: TCNT1 : nilai *timer* (hex)

f_{clk} : Frekuensi *clock* (kristal) yang digunakan (Hz)

T : waktu yang diinginkan/T_{on} atau T_{off} (detik)

N : *prescaler* (1, 8, 64, 256, 1024)

65536 : nilai maksimum *timer*.

2. Pengaturan register TCCR1B, yaitu register yang digunakan untuk mengatur mode *timer* dan *prescaler* yang ditunjukan pada **Tabel 3.9** berikut.

Tabel 3.9. Pengaturan register TCCR1B

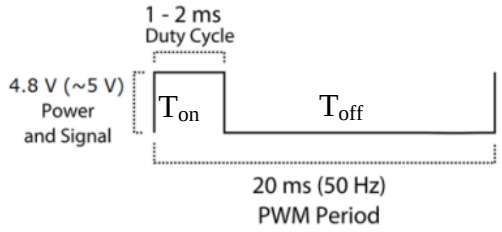
ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10
0	0	0	0	0	0	1	1

Pada **Tabel 3.10** berikut adalah pengaturan 3-bit untuk memilih *prescaler* yang akan digunakan.

Tabel 3.10. Pengaturan 3-bit untuk memilih *prescaler*

CS12:10	**Deskripsi**
001	*No Prescaler*
010	8
011	64
100	256
101	1024

Untuk mengatur PWM pada motor servo, maka dibutuhkan pengaturan T_{on} dan T_{off} yang ditunjukan pada **Gambar 3.29** berikut.



Gambar 3.29. Pengaturan PWM Motor Servo

- Untuk gerakan fleksi tangan robot, untuk semua jarinya membutuhkan nilai

T_{on} dan T_{off} yang sama, yaitu T_{on} 0,6 mS dan T_{off} 20 mS. Pada **Gambar 3.30** berikut adalah hasil gerakan fleksi pada tangan robot.



Gambar 3.30. Hasil Gerakan Fleksi pada Tangan Robot

- Untuk gerakan ekstensi tangan robot juga, masing-masing jari membutuhkan nilai T_{on} dan T_{off} yang berbeda. Nilai tersebut ditunjukan pada **Tabel 3.11** di bawah. Untuk mendapatkan nilai *timer* TCNT1 (T_{on} atau T_{off}) maka digunakan rumus 3.9 sebelumnya. Pada **Gambar 3.31** berikut adalah hasil gerakan ekstensi pada tangan robot.

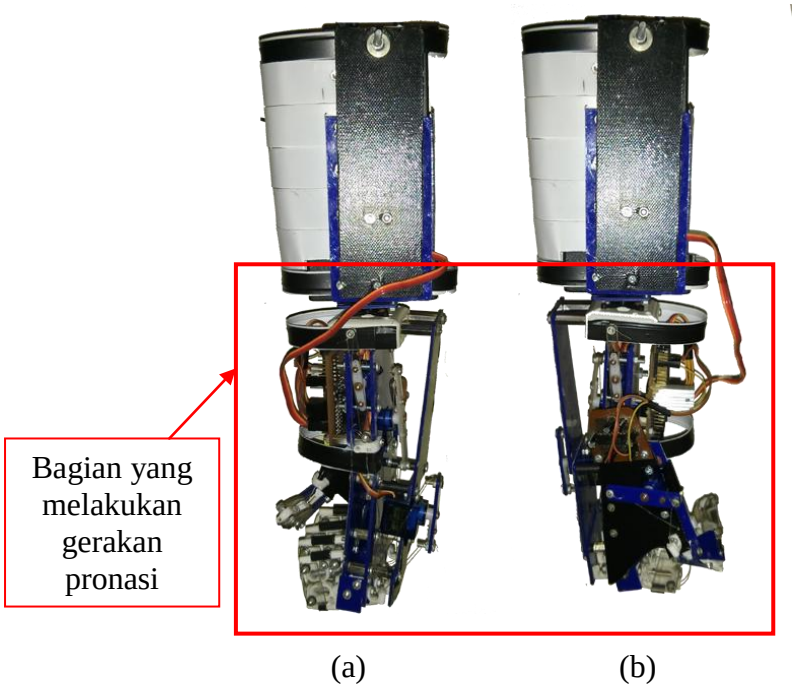
Tabel 3.11. Nilai T_{on} dan T_{off} untuk Gerakan Fleksi pada Tangan Robot

	Kelingking	**Manis**	**Tengah**	**Telunjuk**	**Jempol**
T_{on} (mS)	2,2	2,6	2,3	2,3	1,7
T_{off} (mS)	20				



Gambar 3.31. Hasil Gerakan Ekstensi pada Tangan Robot

- Untuk gerakan pronasi pada lengan tangan robot diberikan nilai PWM dengan T_{on} 0,6 mS dan T_{off} 20 mS. Dan untuk gerakan supinasi pada lengan robot diberikan T_{on} 2,15 mS dan T_{off} 20 mS. Pada **Gambar 3.32** berikut adalah hasil pergerakan pronasi dan supinasi pada lengan robot yang sudah dirancang.



Gambar 3.32. (a) Hasil Gerakan Supinasi Lengan Robot, (b) Hasil Gerakan Pronasi Lengan Robot